

Lab: Function Component Event Planner Landing Page



Estimated time needed: 40 minutes

What you will learn

In this lab, you are going to create a landing page for an organization that plans personal and corporate events. You will create the **EventPlanner** component that consists of various sections, such as the description of the application, event categories, features, testimonials, contact form, and footer. Each section describes various aspects of the Event Planner app, such as the types of events that can be planned, the app's main features, customer reviews, and how to get in touch with the service provider.

You will structure the React component using JSX syntax and use basic HTML elements such as `<div>`, `<header>`, and `<h1>`. Additionally, you will gain insight into the importance of maintaining a clear and organized code structure, as well as the role of component reusability and modularity in building scalable React applications.

Learning objectives

After completing this lab, you will be able to:

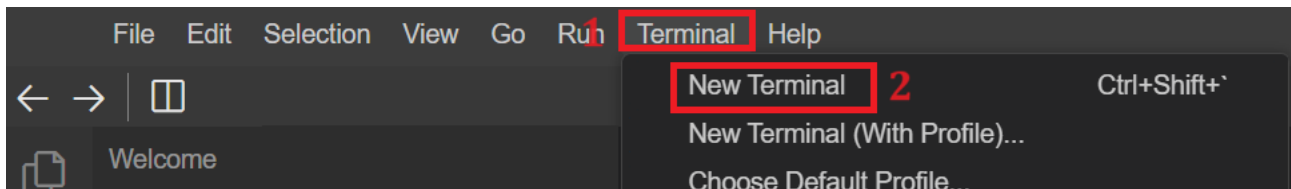
- Create a structured hierarchy of React components to organize the layout of the Event Planner landing page ensuring easy maintenance of issues
- Write the user interface elements, such as headers, paragraphs, lists, and buttons, using JSX code
- Use dynamic content rendering to show testimonials, event categories, and features that are dynamically fetched from data sources
- Use semantic HTML markup to ensure usability and search engine optimization (SEO) benefits

Prerequisites

- Basic Knowledge of HTML
- Intermediate knowledge of JavaScript
- Basic Knowledge of React function component

Setting up the environment

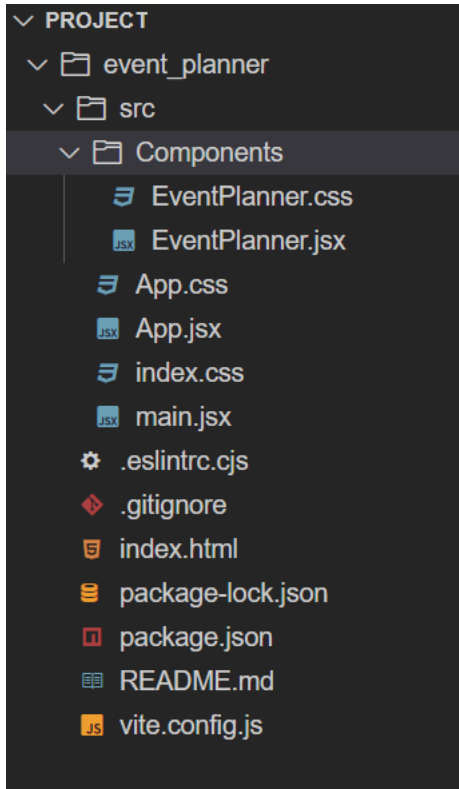
1. From the menu on top of the lab, click on the **Terminal** tab at the top-right of the window shown at number 1 in the given screenshot, and then click **New Terminal** as shown at number 2.



2. Write the following command in the terminal to clone the boiler template for this React application and hit Enter.

```
git clone https://github.com/ibm-developer-skills-network/event_planner.git
```

3. The above command will create a folder named as **event_planner** under **Project** folder and the structure will be as shown in the given screenshot.



4. Write the command to enter the **event_planner** folder in the terminal. Use the below command to navigate to the **event_planner** folder in the terminal.

```
cd event_planner
```


5. To make sure that the code you have cloned is working correctly, you need to follow the given steps:

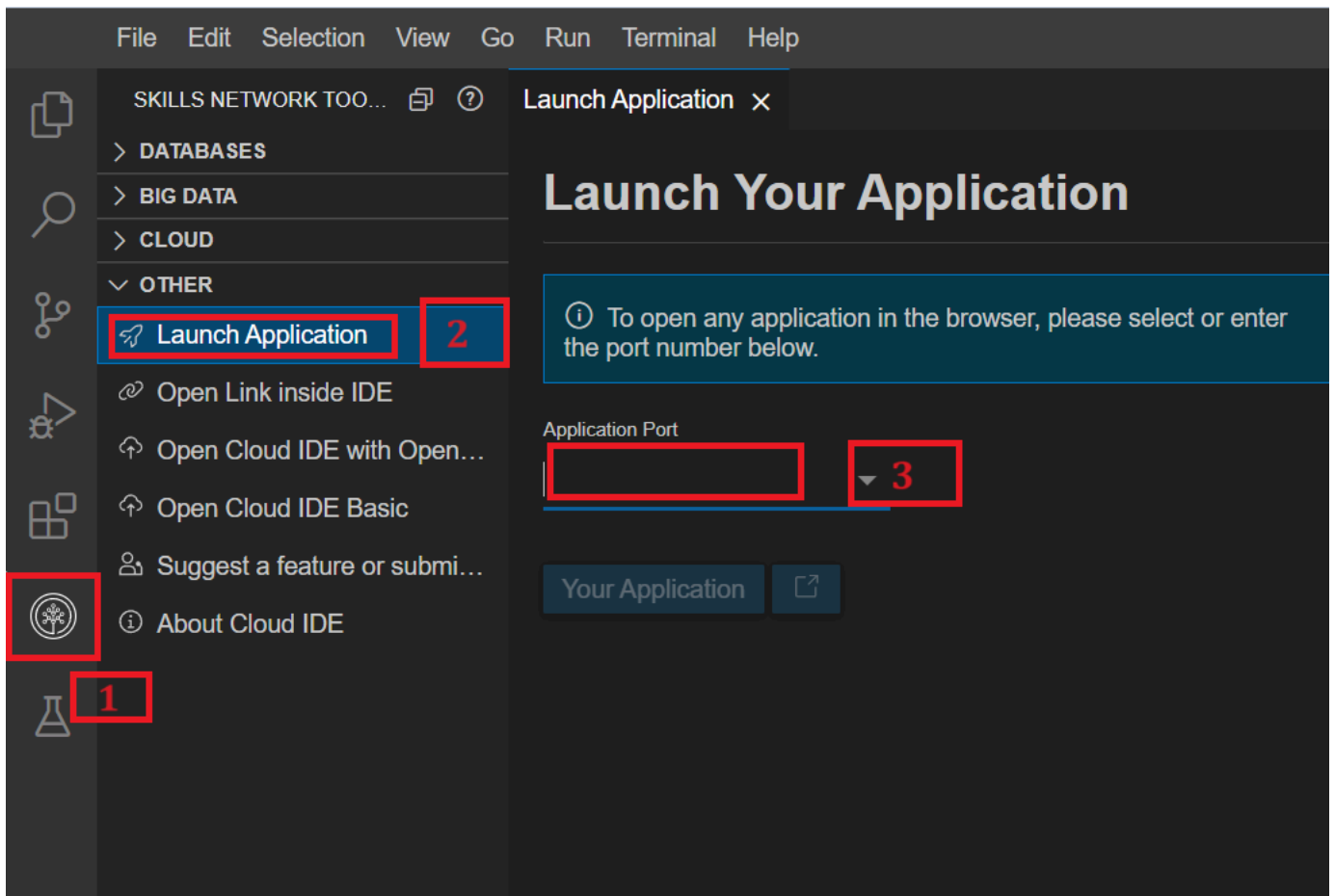
- o Write the following command in the terminal and hit Enter to install all the necessary packages to execute the application:

```
npm install
```

- o Then execute the following command to run the application and this will provide you with port number 4173:

```
npm run preview
```

6. To view your React application, click the Skills Network icon on the left panel (refer to number 1). This action will open the **SKILLS NETWORK TOOLBOX**. Next, click **Launch Application** (refer to number 2). Enter port number **4173** in **Application Port** (refer to number 3) and click .



7. The output will display as shown in the given screenshot.



Create Basic Template

1. Navigate to the **Components** folder located within the src folder of your React project. The event_planner application folder includes the class component named as **EventPlanner.jsx** and a css file named as **EventPlanner.css**.
2. Navigate to the **EventPlanner.jsx** component located within the src folder of your React project. In the return of this component you have a parent `<div>` with class name **event-planner-container** which includes child `<header>` tag with `<h1>` tag.
3. You need to create five `<section>` tags after the `<header>` tag.

- o First section tag with class name `description`
- o Second section tag with class name `events_categories`
- o Third section tag with class name `features`
- o Fourth section tag with class name `testimonials`
- o Fifth section tag with class name `contact`

```
<div className="event-planner-container">
  { /* Page Header */ }
  <header>
    <h1>Welcome to Event Planner</h1>
  </header>
  { /* Section for describing the purpose or overview of the app */ }
  <section className="description">
    { /* Description content goes here */ }
  </section>
  { /* Section to list or categorize different types of events */ }
  <section className="events_categories">
    { /* Event categories content goes here */ }
  </section>
  { /* Section to highlight app features or functionalities */ }
  <section className="features">
    { /* Features content goes here */ }
  </section>
  { /* Section to showcase user reviews or testimonials */ }
  <section className="testimonials">
    { /* Testimonials content goes here */ }
  </section>
  { /* Section to provide contact information or a contact form */ }
  <section className="contact">
```

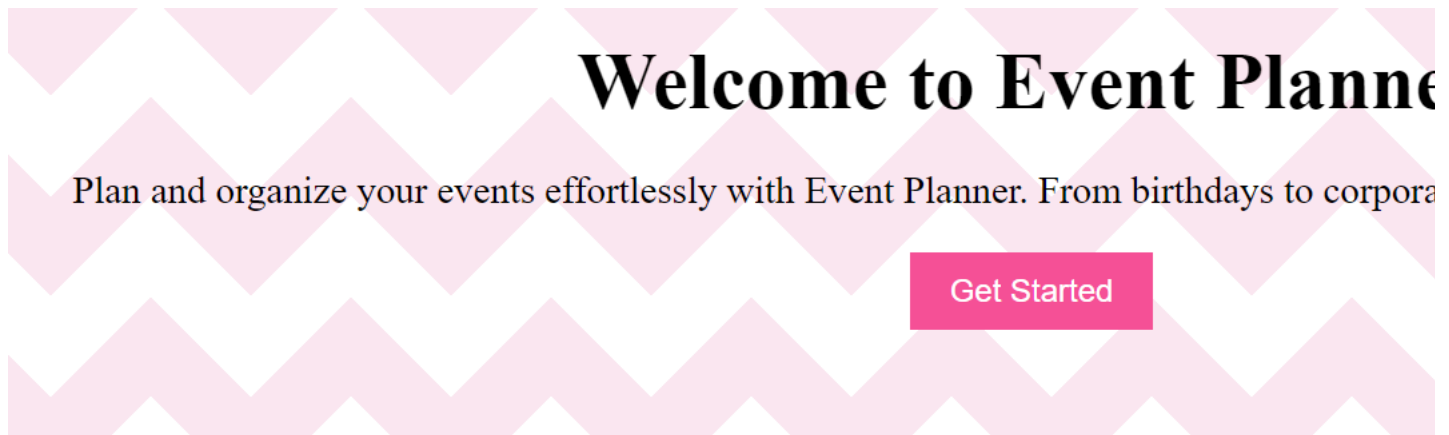
```
        {/* Contact content goes here */}  
      </section>  
</div>
```

Define Description and Event Categories

1. Inside the section tag with class name **description**, add a `<p>` tag and a button tag. In the `<p>` tag, insert text related to the event planning theme. Assign the class name **get-started-button** to the button tag.

```
<section className="description">  
  {/* Brief introduction or marketing message */}  
  <p>  
    Plan and organize your events effortlessly with Event Planner. From  
    birthdays to corporate meetings, we've got you covered.  
  </p>  
  {/* Primary call-to-action button */}  
  <button className="get-started-button">Get Started</button>  
</section>
```

2. Now you need to rerun the React application and see the output. The output should be as per the given screenshot.



Note: If the page is already open in the browser, you will need to refresh it

3. Next you need to create multiple events categories inside section tag with class name **events_categories**. For this create `` tag inside this section. Inside `` tag you should have a `<h2>` tag for event heading and multiple `` tags for different categories of that particular event.

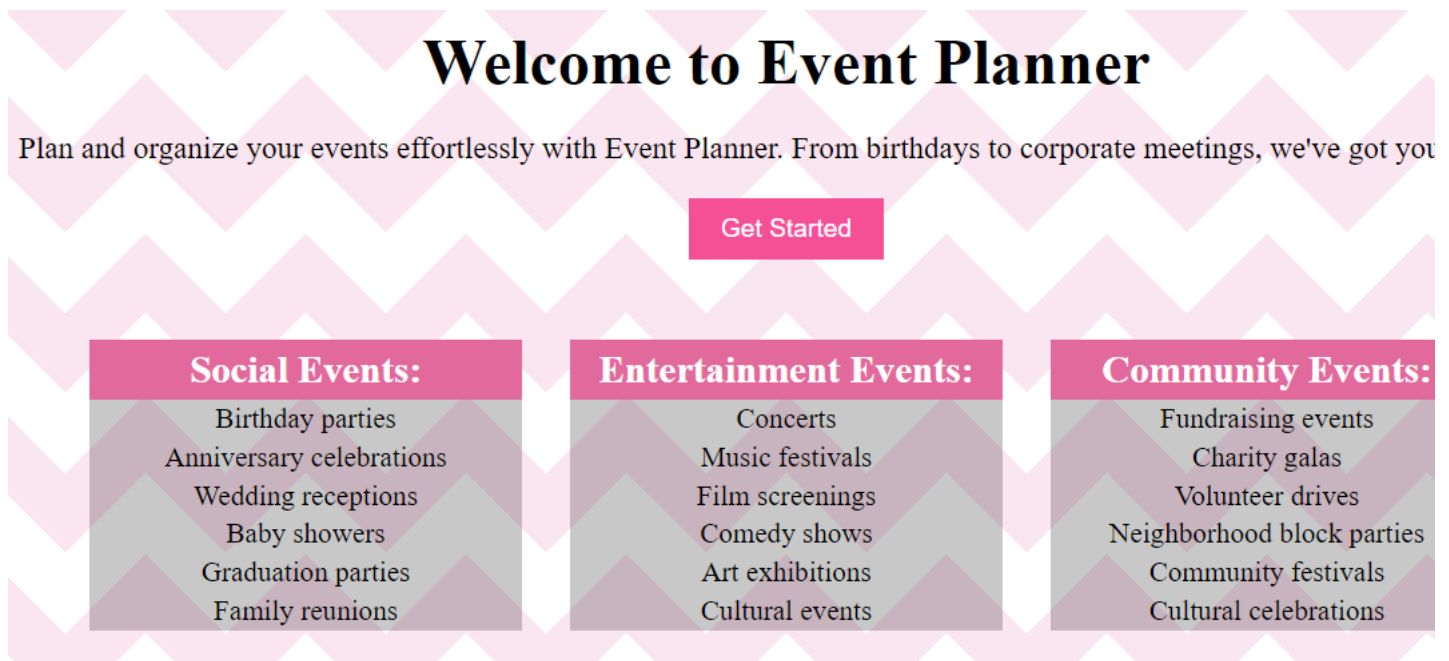
```
<ul>  
  <h2>Social Events:</h2>  
  <li>Birthday parties</li>  
  <li>Anniversary celebrations</li>  
  <li>Wedding receptions</li>  
  <li>Baby showers</li>  
  <li>Graduation parties</li>  
  <li>Family reunions</li>  
</ul>
```

4. You can also include multiple `` tags for different event headings and sub categories just as above.

```
<section className="events_categories">  
  {/* Social event types */}  
  <ul>  
    <h2>Social Events:</h2>  
    <li>Birthday parties</li>  
    <li>Anniversary celebrations</li>  
    <li>Wedding receptions</li>  
    <li>Baby showers</li>  
    <li>Graduation parties</li>  
    <li>Family reunions</li>
```

```
</ul>
{/* Entertainment-based event types */}
<ul>
  <h2>Entertainment Events:</h2>
  <li>Concerts</li>
  <li>Music festivals</li>
  <li>Film screenings</li>
  <li>Comedy shows</li>
  <li>Art exhibitions</li>
  <li>Cultural events</li>
</ul>
{/* Community-focused event types */}
<ul>
  <h2>Community Events:</h2>
  <li>Fundraising events</li>
  <li>Charity galas</li>
  <li>Volunteer drives</li>
  <li>Neighborhood block parties</li>
  <li>Community festivals</li>
  <li>Cultural celebrations</li>
</ul>
</section>
```

5. Again re-run the application and check the output which will display as shown in the given screenshot.



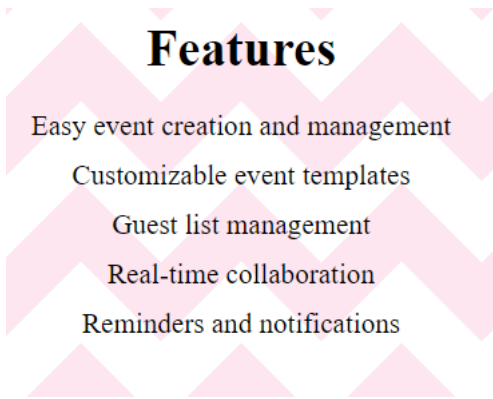
Note: If the page is already open in the browser, you will need to refresh it

Create Features Section

1. Now you need to display the different features that this event planner organization is providing services for.
2. Inside section tag with class name **features** create a **<h1>** tag for heading and one **** tag with multiple **** tags to demonstrate various features offered for event planning.

```
<section className="features">
  {/* Section heading */}
  <h2>Features</h2>
  {/* List of key platform features */}
  <ul>
    <li>Easy event creation and management</li>
    <li>Customizable event templates</li>
    <li>Guest list management</li>
    <li>Real-time collaboration</li>
    <li>Reminders and notifications</li>
  </ul>
</section>
```

3. The above code will display the output as the given screenshot after events categories.



4. To view the latest updates, you must rerun the application, and if the page is already open in the browser, you will need to refresh it.

Create Testimonials Section

1. To create the testimonials section given by multiple users, you need to create `<h1>` tag and `<div>` tag inside section tag with class name **testimonials**.
2. Then inside the `<h2>` tag you need to give the main heading of this section and inside the `<div>` tag you can write the review given by the user along with the username.

```
<section className="testimonials">
  <h2>Testimonials</h2>
  <div className="testimonial">
    <p>"Event Planner made organizing my wedding a breeze. Highly recommended!"</p>
    <p className="author">- Emily Johnson</p>
  </div>
</section>
```

3. You can also create mutiple testimonials like above.

```
<section className="testimonials">
  { /* Section heading */ }
  <h2>Testimonials</h2>
  { /* Individual testimonial block */ }
  <div className="testimonial">
    <p>"Event Planner made organizing my wedding a breeze. Highly recommended!"</p>
    <p className="author">- Emily Johnson</p>
  </div>
  { /* Another testimonial block */ }
  <div className="testimonial">
    <p>"I use Event Planner for all my corporate events. It saves me so much time and effort!"</p>
    <p className="author">- John Smith</p>
  </div>
</section>
```

4. The output will display as shown in the given screenshot after the **Features** section.



Create Contact Section

1. Now, create a contact section so that any visitor to the webpage can easily reach out to the organization's employees.
2. For this again you can have one `<h2>` tag and one `<form>` tag with multiple input boxes and a `<button>` tag inside section tag with class name **contact**.

```
<section className="contact">
  {/* Section heading */}
  <h2>Contact Us</h2>
  {/* Contact form */}
  <form>
    {/* Name input field */}
    <input type="text" placeholder="Name" />
    {/* Email input field */}
    <input type="email" placeholder="Email" />
    {/* Message textarea */}
    <textarea placeholder="Message"></textarea>
    {/* Submit button */}
    <button className="submit-button">Send</button>
  </form>
</section>
```

3. Now re-run the application again in the terminal and check the output. The output will display as shown in the screenshot.

Contact Us

Name

Email

Message

Send

4. ► [Click here for the entire code of EventPlanner.jsx component](#)

Practice Exercise

1. In this practice exercise you will create one footer component which will be included in the **EventPlanner.jsx** component. With the help of this you will understand the concept of composition if components as well.
2. For this you need to right click on **Components** folder after selecting it and select **New File**. Give this new file name **Footer.jsx**. This will create a new component inside **Components** folder.
3. Inside **Footer.jsx** component, first create basic layout for the function component.

Hint: Use function component boiler plate to create this layout.

- [Click here for the solution](#)

4. create one `<footer>` tag and you can include any information you want to display about the organization.

Hint: Use footer tag and include multiple tags such as `<p>`, `<div>` or `<section>` tag.

- [Click here for the solution](#)

5. Include **Footer.jsx** component as last tag of div with class name **event-planner-container** within **EventPlanner.jsx** component.

Hint: Use import keyword to include **Footer** component as child at the top in parent **EventPlanner** component.

- [Click here for the solution](#)

Contact Us

© Event Planner Organization. All rights reserved.

Note: To see the latest changes you need to execute `npm run preview` again in the terminal.

7. ▶ [Click here for the entire code of EventPlanner.jsx component](#)
8. ▶ [Click here for the entire code of Footer.jsx component](#)

Conclusion

Congratulations! You have created your third React application!

- In this lab, you have learned to create a structured hierarchy of React components by creating a component called EventPlanner.
- In addition, you learned how to create forms in React by adding a contact form to a component. This shows them how to combine form elements like textareas and input fields with a send button.
- You used static content like titles, paragraphs, and list items in this component to give information about the event planning app and its features.

Author(s)

Richa Arora

© IBM Corporation. All rights reserved.